الأحداث

نتكلم عن بعض الإحداث الأكثر استخداما :

١. أحداث التركيز

عندما تستقبل الأداة التركيز يتم تنفيذ الحدث FocusGained وعندما مغادرة التركيز من الأداة يتم تنفيذ الحدث FocusLost

٢. أحداث الفأرة Mouse Event

هي عبارة عن مجموعة الأحداث :

- أحداث الفأرة Mouse Events

1) Mouse Events
   - a mouse button is pressed
   - a mouse button is released
   - a mouse button is clicked (pressed and released)
   - the mouse cursor enters a component عند دخول الفارة للأداة
   - the mouse cursor exits a component عند خروج الفارة من الأداة
2) Mouse Motion Events
   - the mouse is moved
   - the mouse is dragged عند السحب داخل الأداة

مثال لمعرفة قيمة إحداثيات النقطة (X,Y) نستخدم الحدث MouseMoved :

```
private void formMouseMoved(java.awt.event.MouseEvent evt)
{
lable_x.                                                          set-
Text("("+String.valueOf(evt.getX())+","+String.valueOf(evt.getY())+")");
{
```

أو لمعرفة أي من أزرار الفأرة تم ضغطها باستخدام الحدث MousePressed أو MouseClicked باستخدام الدالة ()getButton:

```
private void formMousePressed(java.awt.event.MouseEvent evt)
{
    if(evt.getButton()==1)  زر الفأرة الأيسر
    if(evt.getButton()==2)  زر الفأرة الأوسط
    if(evt.getButton()==3)  زر الفأرة الأيمن
}
```

٣. أحداث المكونات Component Event

تنفذ هذا الأحداث عند تحريك المكون أو تغير حجمه أو إخفاء المكون وغيرها وهي:

- ComponentHidden: تنفذ عند إخفاء العنصر أي عند تغير الخاصية Visible=false .
- ComponentShown: عكس الحدث السابق .
- ComponentMoved: تنفذ عندما يتم تحريك المكون .
- ComponentResized: تنفذ عند تغير حجم المكون .

٤. أحداث النافذة **Window Event**

تنفذ هذه الأحداث عندما تم تغير على حالة النافذة، مثلا عندما تم تنشيط النافذة تم تنفيذ الحدث WindowActivated ، وعند عدم التنشيط يتم تنفيذ الحدث WindowDeactivated ، وكذلك أحداث فتح وغلق النافذة WindowOpened WindowClosed والحدث WindowClosing يتم تنفيذه قبل الحدث WindowClosed

٥. أحداث لوحة المفاتيح

تنفذ هذه الأحداث عند الضغط على لوحة المفاتيح وهي :

- KeyTyped : ينفذ عند عملية النقر على أي مفتاح في لوحة المفاتيح ماعدا بعض الأحرف التي تقوم بوظيفة معينة مثل (Home, Page Up, Page Down, Print Screen, Caps Lock) .
- KeyPressed: ينفذ عند عملية النقر على أي مفتاح في لوحة المفاتيح و يتم تنفيذ هذا الحدث بعد الحدث KeyReleased أو الحدث KeyTyped.
- KeyReleased: يتم تنفذ الحدث عند تحرير عملية الضغط .

٦. الحدث Item Event

يعتبر من الأحداث الذكية حيث يمكن من خلاله تحديد العناصر التي تم اختيارها ويولد هذا الحدث لدي المكونات القابلة للاختيار.

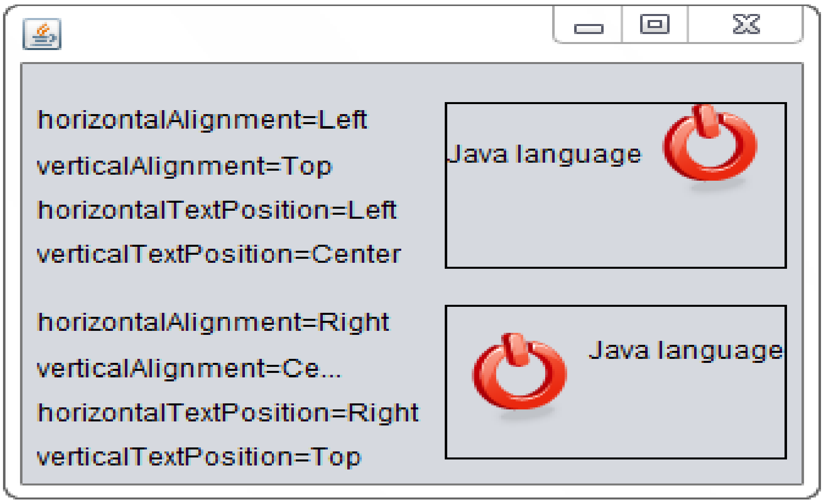
بعض أدوات التحكم في الجافا باستخدام برنامج الـ NetBeans

١. أداة العنوان أو التسمية Jlable

تستخدم هذه الأداة لتوضيح معلومات عند تصميم الواجهات حتى يتسنى للمستخدم التعرف على الواجهة من خلال التوضيح بالنص أو بالصورة، أي أن المستخدم لن يتمكن من التعديل على النص المسند لهذه الأداة، وتحتوي هذا الأداة على مجموعة من الخصائص منها :

Background	لون الخلفية لهذا العنصر
font	نوع الخط و الحجم
foreground	لون الخط
horizontalAlignment	محاذاة النص أو الصورة أفقياً
verticalAlignment	محاذاة النص أو الصورة عمودياً
text	النص التوضيحي
border	وهي عبارة عن الإطار بالنسبة للأداة
insets	وتحدد اقل قيمة للمسافة حول الأداة
Enabled	تمكين استخدام الأداة أو لا
focusable	هل يمكن نقل التركيز إلى الأداة أو لا بمعنى تصبح للقراءة فقط
name	وهي عبارة عن الاسم البرمجي للأداة
icon	وضع صورة توضيحية
horizontalTextPosition	محاذاة النص بالنسبة للصورة افقياً
verticalTextPosition	محاذاة النص بالنسبة للصورة عمودياً

مثال1: يوضح بعض خصائص الأداة يتم تطبيقه في المعمل كما في الصورة:



٢. الأداة JTextField

تستخدم هذه الأداة للحصول على سطر واحد من المعلومات المدخلة بواسطة المستخدم أو من خلال عرض سطر من المعلومات مثال الاسم الأول لشخص أو العنوان السكني له، نذكر بعض خصائص لهذه الأداة وسوف نتجاهل الخصائص المكررة هي:

text	تحتوي على المعلومات المدخلة من قبل المستخدم
editable	إمكانية التعديل على النص
caretColor	لون مؤشر الكتابة
caretPosition	موقع مؤشر الكتابة و تعتمد على طول السلسلة
margin	تحديد الهامش من الجهات الأربع
SelectedTextColor	لون الخط المضلل
selectionColor	لون تضليل النص
SelectionStart	قيمة موقع بدء التضليل
SelectionEnd	قيمة موقع نهاية التضليل

٣. الأداة Text Area

تستخدم هذه الأداة للحصول على سطور متعددة من المعلومات المدخلة بواسطة المستخدم أو العرض مثال عندما تريد كتابة تعليق أو ملاحظة ، في الأسفل بعض الخصائص لهذه الأداة (يوجد خصائص مشتركة مع الأداة السابقة لن نذكرها هنا) :

TabSize	كم عدد الحروف الذي تم تجاوزها عند النقر على tab
LineCount	عدد السطور

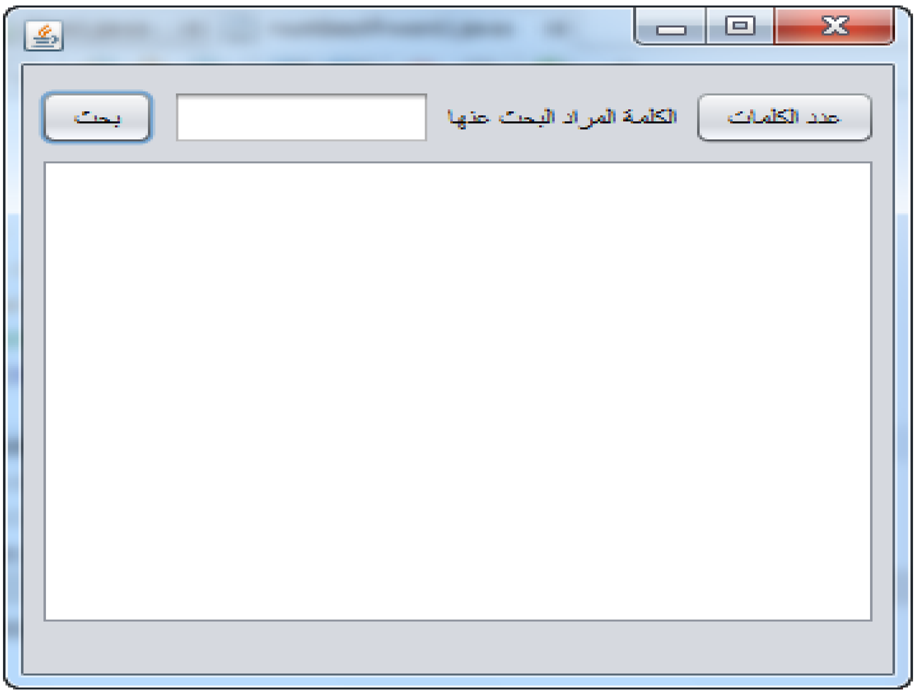
٤. الأداة Button

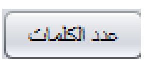
تولد هذه الأداة حدث يقوم بتنفيذ أمراً ما عندما يقوم المستخدم بالنقر على الأداة ، مثال إذا قمت بتصميم برنامج لمعالجة بيانات المرضى بعد أن يتم إدخال البيانات المطلوبة استخدام هذه الأداة لحفظ البيانات.

مثال2 : تصميم برنامج يقوم بإيجاد عد الكلمات و عدد الحروف و عدد الأسطر في سلسلة معينة وكذلك يقوم بالبحث عن كلمة معينة داخل السلسلة.

الإجابة:

شكل الواجهة سيتم ترتيبها كما في الصورة التالية :



في الحدث MouseClicked التابع للأداة 

نستدعي الدالة التالية :

```
private void numbofword(String text)
{
    int index=0,numWord=0,space=0;
    while(index<text.length())
    {
      //     تكرار تجاوز أحرف الكلمة حتى نصل إلى نهاية الكلمة
       while(index<text.length() && text.charAt(index)!=' ')
       index++;
       if(index!=0)
       numWord++;
      //    تكرار تجاوز وحساب المسافات
       while(index<text.length()&& text.charAt(index)==' ')
       {
         index++;
         space++;
       }
    }
String message="عدد الكلمات هي      "+numWord+"\n";
message+="عدد الأحرف بدون المسافات  "+(text.length()-space)+"\n";
message+="عدد الأحرف مع المسافات    "+text.length()+"\n";
message+="عدد الأسطر    "+jTextArea1.getLineCount()+"\n";
JOptionPane.showMessageDialog(null, message);
}
```

الفكرة هي نستخدم تكرر للمرور على كل حرف النص و تكرار يتجاوز أحرف الكلمة حتى نصل إلى مسافة نكون وصلنا نهاية الكلمة نزيد المتغير (numWord) بواحد وبعد ذلك

تكرار اخر يحسب عدد المسافات ويعمل عمليات تجاوز لها حتى ننتقل إلى بداية الكلمة الأخرى طبعا كلا التكرارات نستخدم فيها دليل واحد. تبقى معنا الشرط if(index!=0) حتى نتجاوز الفراغات التي في الأمام لان تكرار تجاوز الأحرف ما رح يحقق الشرط وبالتالي ما رح يزيد قيمة الدليل (index=0).

في الحدث MouseClicked التابع للأداة 

نستدعي الدالة التالية :

```
private boolean searchword(String text , String word)
{
    String[] split=text.split(" ");
    for(String word2:split)
      {
         g+=word2.length()+1;
         if(word2.equals(word))
           return true;
      }
     return false;
}
```

الفكرة هي تقسيم النص إلى جمل باستخدام الدالة ;split(" ") ثم تم استخدام تكرار للمرور على الكلمات ومقارنتها مع الكمة المراد البحث عنها. الدالة تستقبل وسيطين الأول النص والثاني الكلمة المراد البحث عنها وترجع قيمة منطقية true أو false .

٥. الأداة JComboBox

وهذه الأداة عبارة عن اتحاد بين أداة الزر وأداة القائمة المنسدلة حيث يمكن للمستخدم أن يختار القيمة من القائمة المنسدلة التي تظهر عند يقوم المستخدم بالنقر على الزر، يمكن الحصول على قيمة العنصر المختار من الحدث ItemStateChanged ولكن سكون التغير هنا بحالتين Deselected و Selected أي سيتم تنفيذ الحدث السابق مرتين بالترتيب، لذلك يجب تحديد متى تريد تنفيذ الحدث كالتالي :

```
if(evt.getStateChange()==ItemEvent.SELECTED)
{
   // your code
}
```

الدالة ()evt.getStateChange ترجع متغير صحيح ، ترجع القيمة واحد دليل على أن الحدث Selected وترجع القيمة 2 على أن الحدث Deselected .

مثال : اكتب برنامج لتحويل بين وحدات الطول ؟

٦. الأداة jCheckBox

تمكن المستخدم من حرية الاختيار عدة خيارات عن طريق تحديد أو إلغاء التحديد على مربع الاختيار الذي توفرها هذه الأداة ، عند الاختيار تم التغير على الخاصية Selected وكذلك يمكنك معرفة هل تم اختيار الخيار أم لا من خلال الدالة ()isSelected وإذا كان هناك عدة خيارات يمكنك استخدام الحدث itemStateChange.

٧. الأداة jRadioButton

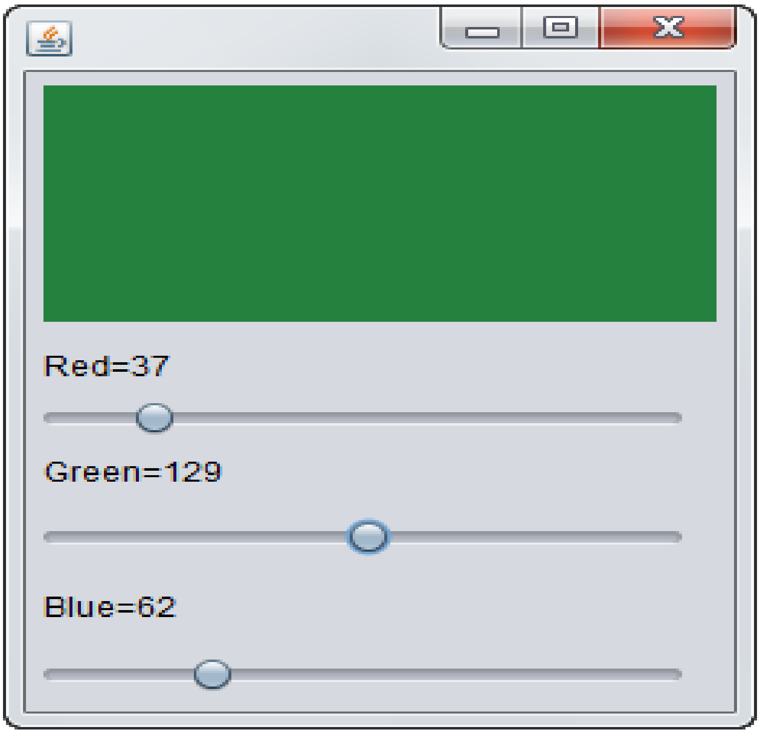
تمكن المستخدم من اختيار خيار واحد من عدة خيارات عند إضافتها على شكل مجموعة ButtonGroup ، ولكن في حال عدم إضافتها على شكل مجموعة تكون مشابهة للأداة jCheckBox.

٨. الأداة jSlider

وهي أداة الانزلاق تسمح للمستخدم باختيار قيمة بواسطة انزلاق العقدة الموجودة داخل حدود الأداة ويمكن تحديد أعلى قيمة يمكن تحديدها واقل قيمة تمكن تحديدها من خلال الخاصيتين maximum و minimum أما الخاصية orientation تحدد اتجاه الانزلاق هل أفقي أم عمودي ، كما يمكن الحصول على القيمة التي تم تحديدها من خلال الدالة ()getValue.

مثال3 : تصميم برنامج يمكن للمستخدم من خلاله عمل دمج للألوان الثلاثة الرئيسية الأحمر والأخضر و الأزرق ؟

الإجابة : سنستخدم ثلاث أدوات من الأداة jSlider وأيضا أداة jpenal واحدة حتى نضع اللون الناتج من عملية الدمج للألوان . قم بتصميم النافذة كما في الصورة .



في الحدث StateChanged التابع للأداة اللون الأحمر jSlider نكتب الكود التالي :

```
jLabel2.setText("Red="+jSlider1.getValue());

jPanel1.setBackground(new
java.awt.Color(jSlider1.getValue(),jSlider2.getValue(), jSlider3.getValue()));
```

كرر الخطوات السابقة على بقية الألوان.

٩. الأداة jFormattedText

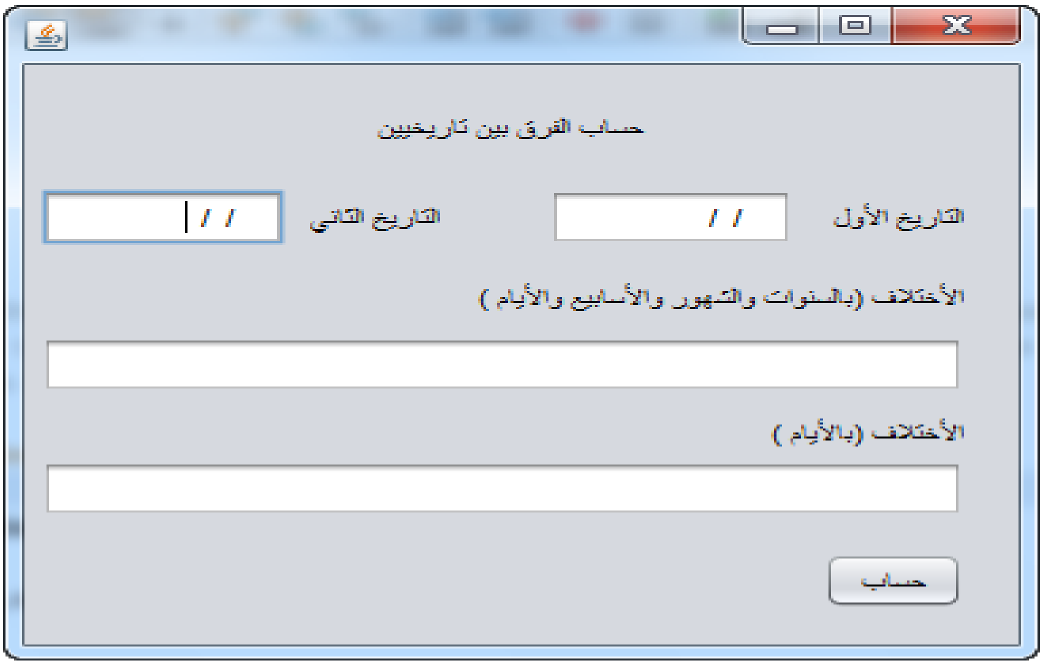
يمكن من خلال هذه الأدا ة التعديل على صيغة القيمة النصية المدخلة من قبل المستخدم وإلزامه بصيغة محدد مثال (التاريخ ، الوقت ، العملة) . الجدول التالي يوضح رموز المستخدمة لاختبار قبول النص :

#	رمز للأرقام
A	رمز للأحرف والأرقام
*	يرمز لأن يكون المدخل أي شيء
U	يرمز للأحرف الكبيرة
L	يرمز للأحرف الصغيرة
H	يرمز لان يكون الإدخال بالنظام السداسي

مثال لكتابة صيغة لرقم التلفون ####-### (###) .

مثال4 : تصميم برنامج لإيجاد الفرق بين تاريخين بالسنين و الشهور والأيام؟

الحل : الأدوات المستخدمة هي أداتين jFormattedText مع تغير صيغة الإدخال إلى تاريخ من خلال الخاصية FormatterFactory ، وكذلك أداتين JTextField لعرض الاختلاف وكذلك أداة العنوان للتوضيح كما هو واضح في الصورة التالية:



في الحدث Click التابع للزر حساب نكتب الكود التالي :

```
long diff = Date.parse(jFormattedTextField2.getText())-
Date.parse(jFormattedTextField1.getText());
```

```
jTextField2.setText(String.valueOf((diff/(1000 * 60 * 60 * 24))));

year_months_day((diff/(1000 * 60 * 60 * 24)));
```

الدالة () parse تحول التاريخ إلى أجزاء من الألف من الثانية وبالتالي نحتاج إلى تحويل جزء الثانية إلى ثانية لذلك نقسم على ألف ثم نحول الثانية إلى دقائق نقسم على 60 ثم إلى ساعة نقسم على 60 أيضاً أخيراً إلى أيام نقسم على 24، أما بالنسبة إلى الدالة year_months_day تقوم بتحويل الأيام إلى سنوات وشهور وأسابيع وأيام :

```
private void year_months_day(long days)
  {
       int years=0,months=0,weeks=0,day=0;
       years=(int)days/365;
       days=days%365;
       months=(int)days/30;
       days=days%30;
       weeks=(int)days/7;
       days=days%7;
       String text=null;
       if(years==1)
         text=+years+" سنة";
       else
         if(years!=0)
```

```
        text=+years+" سنوات";
    text+=" و "+months+" شهر/شهور";
    if(weeks!=0)
    text+=" و "+weeks+" اسبوع";
    text+=days+"يوم/ايام ";
   jTextField1.setText(text);
}
```

تم تعريف ثلاث متغيرات صحيحة لكلاً من السنة، والشهر و الأسبوع الفكرة هي عندما نقسم الأيام days على 365 يوم وهي عدد أيام السنة سكون هناك باقي يمثل عدد الأيام التي هي اقل من سنة وهي شهور حتى نوضح أكثر نفرض أن days=395:

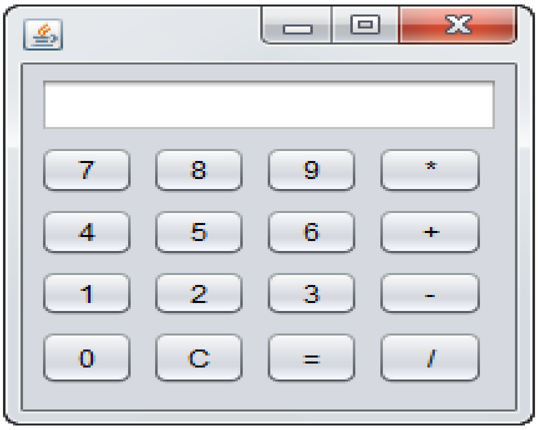
years=(int)days/365 ⟶ years=(int) 1.08219 ⟶ years=1

days=days%365 ⟶ years=(int)30 ⟶ years=30

وهكذا بالنسبة للشهور والأسابيع، طبعاً حتى تتفادى ذكر 0 سنة، شهر، أسبوع أو يوم اختبر هل المتغيرات لا تساوي الصفر.

مثال5 : تصميم برنامج حاسبة تقوم بعمليات الجمع والطرح والضرب والقسمة بحيث يستقبل الأرقام أو العمليات من خلال الأزرار أو من خلال لوحة المفاتيح ؟

الحل: قم بتصميم الواجهة كما في الصورة التالية :



في الحدث ActionPerformed التابع للأزرار الخاصة بالأرقام نكتب الكود التالي حسب الرقم الخاص للأزر مثلاً الزر رقم 5 :

```
jTextField2.setText(jTextField2.getText()+"5");
```

أزار العمليات مثلاً زر عملية الجمع :

```
x=Integer.parseInt(jTextField2.getText());
operation='+';
isset=true;
```

المتغير x هو متغير double ويحتوي على العدد الأول، والمتغير operation يحتوي على نوع العملية اما المتغير isset حتى نعرف انه تم النقر على نوع العملية لبيداء إسناد الرقم الثاني و حتى لا يتم مسح الرقم الأول إلا عند كتابة الرقم الثاني، كرر الخطوات السابقة لجميع العمليات .

زر المساواة نكتب الكود التالي :

```
try
    {
     y=Double.parseDouble(jTextField2.getText());
```

```
    if(operation=='-')
        jTextField2.setText(String.valueOf(x-y));
    else
      if(operation=='+')
          jTextField2.setText(String.valueOf(x+y));
      else
        if(operation=='*')
            jTextField2.setText(String.valueOf(x*y));
        else
        {
          if(operation=='/' && y!=0)
              jTextField2.setText(String.valueOf(x/y));
          else
              JOptionPane.showMessageDialog(null, "error divion");
        }
     isset=false;
   }
   catch(Exception ee)
   {        }
```

إلى هنا يكون الحساب الأزرار، أما بالنسبة للإدخال من لوحة المفاتيح نستخدم الحدث KeyPressed ولكن سنستخدم هذا الحدث في أي أداة لأن عندما يكون التركيز على أداة أخرى لن ينفذ الحدث لذا نقوم بتعطيل الخاصية focusable لجميع الأدوات ماعدا النافذة Form حتى تصبح النافذة دائماً في وضع التركيز وفي الحدث KeyPressed التابع للنافذة نكتب الكود التالية :

وهذا الكود إسناد الأرقام من لوحة المفاتيح :

```
if((evt.getKeyCode()>=48  &&  evt.getKeyCode()<=57)||
(evt.getKeyCode()>=96 && evt.getKeyCode()<=105)  )

      {

         if(isset)

         {

            jTextField2.setText(null);

            isset=false;

         }

      jText-
Field2.setText(jTextField2.getText()+String.valueOf(evt.getKeyChar()));

      }
```

والكود التالي نستدعى فيه أحداث أزرار العمليات :

```
 if(evt.getKeyCode()==45 ||evt.getKeyCode()==109 )

      {

         jButton3ActionPerformed(null);
```

```
}
else
    if(evt.getKeyCode()==107)
    {
        jButton12ActionPerformed(null);
    }
    else
        if(evt.getKeyCode()==106)
        {
            jButton13ActionPerformed(null);
        }
        else
            if(evt.getKeyCode()==111)
            {
                jButton14ActionPerformed(null);
            }
 if(evt.getKeyCode()==61 || evt.getKeyCode()==10)
{
   jButton4ActionPerformed(null);
```

```
}
```

عند تعطيل الخاصية focusable لن تستطيع تنفيذ backspace لذا نكتب الكود التالي لعمل مسح :

```
String s=jTextField2.getText();
        try
        {
           if(evt.getKeyCode()==8)
             jTextField2.setText(jTextField2.getText(0,s.length()-1));
        }
        catch(Exception ee){}
```

I hope you have all the best